Tarea #3

Jesús Enrique Librado Polanco
A01551367
Diseño de compiladores

April 20, 2020

Problema

$$
\begin{array}{lcl}
prog & \rightarrow & \textbf{program id } opt\_stmts \\
\\
stmt & \rightarrow & \textbf{set id } expr \\
 & | & \textbf{if } (expresion)\ opt\_stmts \\
 & | & \textbf{ifelse } (expresion)\ opt\_stmts\ opt\_stmts \\
 & | & \textbf{while } (expresion)\ opt\_stmts \\
\\
opt\_stmts & \rightarrow & \textbf{\{ \}} | \textbf{\{ } stmt\_lst \textbf{ \}} \mid instr \\
\\
stmt\_lst & \rightarrow & instr \mid stmt\_lst\ instr \\
\\
instr & \rightarrow & \textbf{ ; } \mid stmt \textbf{ ;} \\
\\
expr & \rightarrow & expr + term \\
 & | & expr \textbf{ - } term \\
 & | & term \\
\\
term & \rightarrow & term \textbf{ * } factor \\
 & | & term \textbf{ / } factor \\
 & | & factor \\
\\
factor & \rightarrow & \textbf{(} expr \textbf{)} \\
 & | & \textbf{id} \\
 & | & \textbf{num} \\
\\
expresion & \rightarrow & expr < expr \\
 & | & expr > expr \\
 & | & expr = expr
\end{array}
$$

Eliminación de la recursión de la izquierda

$$
\begin{array}{lcl}
prog & \rightarrow & \textbf{program id} \quad opt\_stmts \\
\\
prog' & \rightarrow & prog \ \$ \\
\\
stmt & \rightarrow & \textbf{set id} \quad expr \\
 & | & \textbf{if}(expresion) \quad opt\_stmts \\
 & | & \textbf{ifelse}(expresion) \quad opt\_stmts \quad opt\_stmts \\
 & | & \textbf{while}(expresion) \quad opt\_stmts \\
\\
opt\_stmts & \rightarrow & \textbf{\{\}} \mid \textbf{\{}stmt\_lst\textbf{\}} \mid instr \\
\\
stmt\_lst & \rightarrow & instr \quad stmt\_lst' \\
stmt\_lst' & \rightarrow & instr \quad stmt\_lst' \mid \epsilon \\
\\
instr & \rightarrow & ; \mid stmt \ ; \\
\\
expr & \rightarrow & term \quad expr' \\
\\
expr' & \rightarrow & + \ term \ expr' \\
 & | & \textbf{-} \ term \ expr' \\
 & | & \epsilon \\
\\
term & \rightarrow & factor \quad term' \\
\\
term' & \rightarrow & \textbf{*} \ factor \ term' \\
 & | & / \ factor \ term' \\
 & | & \epsilon \\
\\
factor & \rightarrow & (expr) \\
 & | & \textbf{id} \\
 & | & \textbf{num} \\
\\
expresion & \rightarrow & expr \ \leq \ expr \\
 & | & expr \ \geq \ expr \\
 & | & expr \ = \ expr
\end{array}
$$

FIRST

Non-terminal	terminals
prog	{program}
prog′	{program}
stmt	{set, if, ifelse, while}
opt_stmts	{{, ;, set, if, ifelse, while}
stmt_lst	{;, set, if, ifelse, while}
stmt_lst′	{ ;, set, if, ifelse, while, ϵ}
instr	{ ;, set, if, ifelse, while}
expr	{(, id, num}
expr′	{+, -, ϵ}
term	{(, id, num}
term′	{*, /, ϵ}
factor	{(, id, num}
expresion	{(, id, num}

FOLLOW

Non-terminal	constraints	terminals
prog		{$}
prog′		{$}
stmt		{;}
opt_stmts	FIRST(*opt_stmts*)∪FOLLOW(*prog*)	{{, ;, set, if, ifelse, while, $}
stmt_lst		{}}
stmt_lst′	FOLLOW(*stmt_lst′*)∪FOLLOW(*stmt_lst*)	{}}
instr	FOLLOW(*opt_stmts*)	{{, ;, set, if, ifelse, while, $}
expr		{≤, ≥, =,), ;}
expr′		{≤, ≥, =,), ;}
term	[FIRST(*expr′*)-ϵ]∪FOLLOW(*expr*)	{+, -, ≤, ≥, =,), ; }
term′	FOLLOW(*term*)	{+, -, ≤, ≥, =,), ; }
factor	[FIRST(*term′*)-ϵ]∪FOLLOW(*term*)	{*, /, +, -, ≤, ≥, =,), ;}
expresion		{)}